4T

УДК 004.272

Л.И. Тимченко, И.Д. Ивасюк, Р.В. Макаренко
Государственный экономико-технологический университет транспорта, г. Киев, Украина
timchen@list.ru

Базисные вычислительные структуры и алгоритмы на основе параллельно-иерархического преобразования

В статье рассматривается возможность построения принципиально новых вычислительных структур и алгоритмов. Для решения такой задачи предлагается теория параллельно-иерархического преобразования, ориентированного на достижение максимально возможного алгоритмического и схематического быстродействия.

Введение

Информационно-вычислительная техника подошла к тому критическому моменту, когда теоретические и прикладные исследования сделали очевидным ограничение в применении для решения целого ряда задач последовательно-поточечных арифметических операций в ЭВМ первых пяти поколений.

Перспективным представляются теоретические и прикладные исследования параллельно-иерархических вычислительных структур (ПИВС).

Установить полное согласованное соответствие между потенциальным параллелизмом, заложенным в задаче, и его реализацией в реальной вычислительной среде – желаемая, но пока труднодостижимая задача. Ведь для этого пришлось бы формировать собственную вычислительную среду (собственную машину) под каждую задачу.

Целью данной работы является исследование принципов построения архитектуры, разработка методов синтеза информационных структур и вычислительных устройств, а также возможность одновременного образного многоэтапного восприятия.

При анализе параллелизма, реализуемого в вычислительных устройствах, выделяют несколько уровней [1]. Это – уровень заданий, алгоритмический, командный и арифметический уровни.

Очевидно, что с точки зрения требований к вычислительной среде, реализующей принцип ПИ преобразования, нас интересуют только те уровни и формы параллелизма, для которых при их реализации необходимо взаимодействие операционных устройств (спецпроцессоров, устройств передачи данных, устройств обработки больших массивов информации) с рассматриваемой вычислительной средой.

Анализ форм параллелизма на последних трех уровнях выполним с использованием терминологии М. Флинна, которая применяется для классификации структур вычислительных устройств [2]. В системе М. Флинна все вычислительные устройства разделяются на четыре класса:

1. ОКОД – одиночный поток команд и одиночный поток данных.
2. ОКМД – одиночный поток команд и множественный поток данных.
3. МКОД – множественный поток команд и одиночный поток данных.
4. МКМД – множественный поток команд и множественный поток данных.

Анализируя структуру вычислительной среды, основанной на методе ПИ преобразования, ее можно отнести к классу МКМД.

Однако в зависимости от степени распараллеливания вычислительного процесса на основе ПИ преобразования возможны различные структуры вычислительной среды, реализующие данный способ, представленные в табл. 1. Структура 1 (табл. 1) осуществляет параллельное преобразование информации и представляет систему класса МКМД. Схематическое представление взаимодействия цифровых автоматов (ЦА) указывает на конвейерную их организацию. Причем данные являются активными элементами вычислительного процесса и используются для инициирования выполнения команд. То есть данная вычислительная структура управляется потоком данных.

Каждый ЦА функционально ориентирован на выполнение конечного, строго определенного числа операций. Информация о том, какая операция должна выполняться цифровым автоматом в данный момент времени, поступает из памяти команд (Пм. К). Команда выполняется тогда, когда данные присутствуют во входных портах ЦА и отсутствуют в выходных портах. Выполнение команды приводит к исчезновению данных во входных портах автомата и появлению результата в выходных портах. Преобразование представляет собой направленный граф, в узлах которого расположены описания действий над данными. Память данных (Пм. Д) хранит результаты прямого преобразования исходных массивов информации и выдает их значения в процессе обратного преобразования. Буферная память принимает для прямого преобразования и передает после обратного преобразования очередной пакет массивов информации. Процесс преобразования информации состоит в «прохождении» через сформированную операционную среду входного потока данных.

Структура 2 (табл. 1) также может быть отнесена к классу МКМД, осуществляющая ПИ преобразование последовательно по уровням, ветви каждого из которых обрабатываются параллельно. Процесс преобразования в такой структуре более продолжительный по времени, однако для ее построения требуются меньшие аппаратурные затраты. Для хранения промежуточного результата при обработке одного уровня в такой структуре используется память для одного уровня (ПОУ). Соответственно каждый ЦА в структуре 2 обрабатывает только одну ветвь.

Структура 3 (табл. 1) отражает последовательный принцип обработки информации по уровням и ветвям, но в пределах каждой ветви обработка происходит параллельно. Данная структура относится к классу ОКМД. Здесь каждый ЦА обрабатывает один элемент ветви.

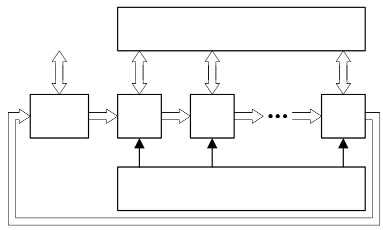
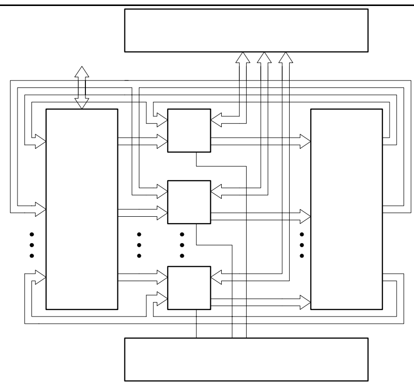
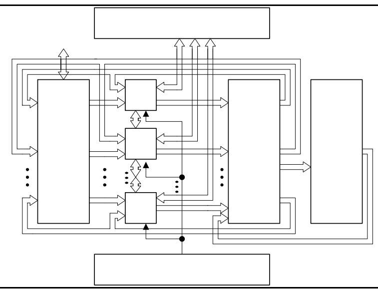
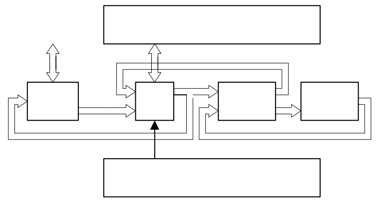
Все автоматы связаны между собой шиной данных и работают с общим устройством управления (УУ). Для хранения промежуточных результатов при обработке ветви предусмотрена соответствующая память. Последняя структура 4 в табл. 1 представляет собой устройство, осуществляющее ПИ преобразование по элементам. Это самая медленноработающая вычислительная структура, но требующая минимальных аппаратурных затрат и совместима с любыми вычислительными структурами и устройствами передачи данных с последовательным принципом организации.

Как показано в [3], преобразование массивов информации на основе метода ПИ преобразования можно производить с различной степенью распараллеливания вычислительного процесса. Это влияет на скорость преобразования информации (время обработки) и на объем используемых вычислительных ресурсов при обработке (аппаратурные затраты).

Естественно, чем выше степень распараллеливания процесса, тем выше быстродействие и больший объем вычислительных средств необходим для кодирования и декодирования (прямого и обратного преобразования). Рассмотрим алгоритмы, отра-

жающие различные уровни распараллеливания процесса ПИ преобразования и различные способы формирования результата преобразования.

Таблица 1 – Представление ПИВС на основе классификации М. Флинна

№ п/п	Метод преобразования и условные обозначения	Структура параллельно-иерархических вычислительных устройств	Форма параллелизма
1	2	3	4
1	Параллельный метод преобразования (параллельное преобразование уровней) Пм.Д. – память данных Пм. К. – память команд П – процессор БОП – блок оперативной памяти на 1 пакет массивов		МКМД
2	Параллельно-последовательный метод преобразования на уровне ветвей (параллельное преобразование ветвей, последовательное преобразование уровней) ПЭ – процессорный элемент БПОУ – блок памяти для одного уровня		МКМД
3	Параллельно-последовательный метод преобразования на уровне элементов (параллельное преобразование одной ветви, последовательное преобразование ветвей и уровней) БПОВ – блок памяти одной ветви		ОКМД
4	Последовательный метод преобразования (последовательное преобразование элементов одной ветви, ветвей одного уровня, уровней) ⇔ – поток данных ↔ – поток команд		ОКОД

Формирование результата при кодировании массивов информации возможно производить несколькими путями. В зависимости от того, какие элементы преобразования используются в качестве результата, будет меняться сходимость процесса, а это, в свою очередь, влечет изменение скорости преобразования и объема выходной информации при кодировании.

Предлагается два наиболее эффективных способа формирования результата:

1. В результат идут только начальные одиночные элементы (первые элементы первых ветвей, кроме первого уровня).

2. В результат идут все одиночные элементы на всех уровнях.

Эти способы предложены вследствие того, что при хорошей сходимости массивов результат кодирования по объему преобразованных данных получается минимальным. Отличие этих способов заключается в том, что при первом способе формирования результата процесс преобразования информации по времени продолжительнее, но результат содержит меньше элементов.

На рис. 1 и рис. 2 представлены блок-схемы алгоритмов параллельного прямого и обратного преобразований. Блоки 2 и 3 (рис. 1.) показывают, что преобразования происходят параллельно по всем уровням. Если во всех четных уровнях выполняется операция выбора по F^* критерию, то во всех нечетных уровнях осуществляется Q^* преобразование, и наоборот.

Участок алгоритма (рис. 1), обведенный пунктирной линией, отражает первый способ формирования результата. Структурные схемы последующих алгоритмов кодирования представлены со вторым способом формирования результата. На процесс декодирования способ формирования результата при кодировании принципиального значения не имеет, поэтому блок-схема алгоритма декодирования не изменяется.

В блоке 4 на рис. 2 анализируется наличие одинаковых элементов, или элементов с последующих уровней по всем уровням. На одних уровнях такие элементы на данном шаге алгоритма могут быть, а на других уровнях отсутствовать. Поэтому из блока 4 возможен выход по двум направлениям одновременно, что противоречит логическому блоку.

Вычислительная сеть, реализующая алгоритм иерархической сетевой параллельной обработки больших массивов информации на основе известных технических средств, достаточно сложна и требует немалых аппаратурных затрат. Поэтому для практических задач целесообразно применять параллельно-последовательный принцип обработки. Ниже рассмотрено несколько алгоритмов преобразования пакетов массивов с различной степенью распараллеливания процесса вплоть до последовательного преобразования каждого массива по элементам.

На рис. 3. и рис. 4 представлены структурные схемы алгоритмов кодирования и декодирования, основанных на параллельной обработке массивов в пределах одного уровня и последовательной обработке самих уровней. Такой способ назовем параллельным преобразованием информации на уровне ветвей одного уровня.

Следующий способ преобразования заключается в параллельной обработке одного массива, который на каждом уровне обрабатывается последовательно. Соответственно последовательно обрабатываются массивы на разных уровнях. Такой способ назовем параллельным преобразованием на уровне элементов одной ветви, структурные схемы алгоритмов кодирования и декодирования которых показаны на рис. 3 и рис. 4.

Последний способ преобразования пакета массивов путем распараллеливания вычислительного процесса заключается в последовательном преобразовании каждого массива по элементам. Назовем его последовательным преобразованием по элементам. Этот способ самый низкопроизводительный, но позволяет осуществлять ПИ преобразование на базе персонального компьютера и применить предложенный метод в научных исследованиях.

Структурные схемы алгоритмов процесса кодирования и декодирования информации на основе последовательного преобразования по элементам показаны на рис. 5 и рис. 6.

Все рассмотренные алгоритмы ПИ преобразования с различной степенью распараллеливания вычислительного процесса на практике необходимо применять в зависимости от типа решаемой задачи и от требуемого на стадии проектирования масштаба времени.

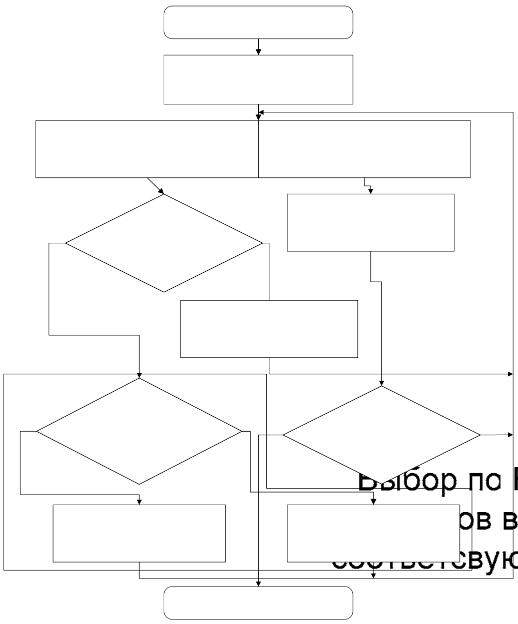


Рисунок 1 – Структурная схема алгоритма прямого ПИ преобразования



Рисунок 2 – Структурная схема алгоритма обратного ПИ преобразования

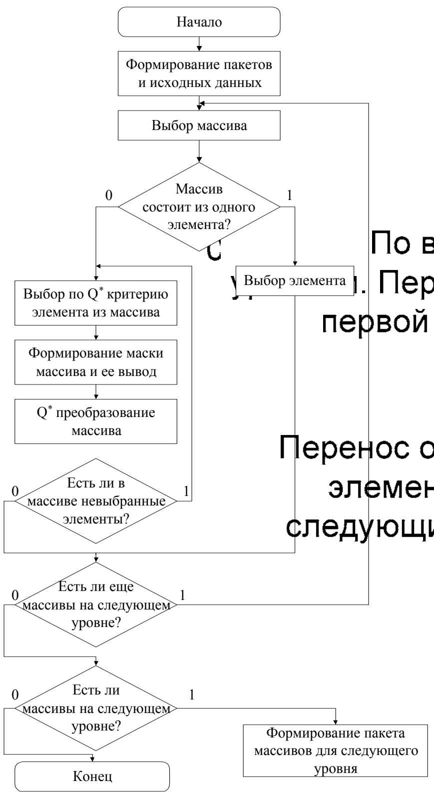


Рисунок 3 – Структурная схема алгоритма прямого ПИ преобразования на уровне элементов одной ветви



Рисунок 4 – Структурная схема алгоритма обратного ПИ преобразования на уровне элементов одной ветви

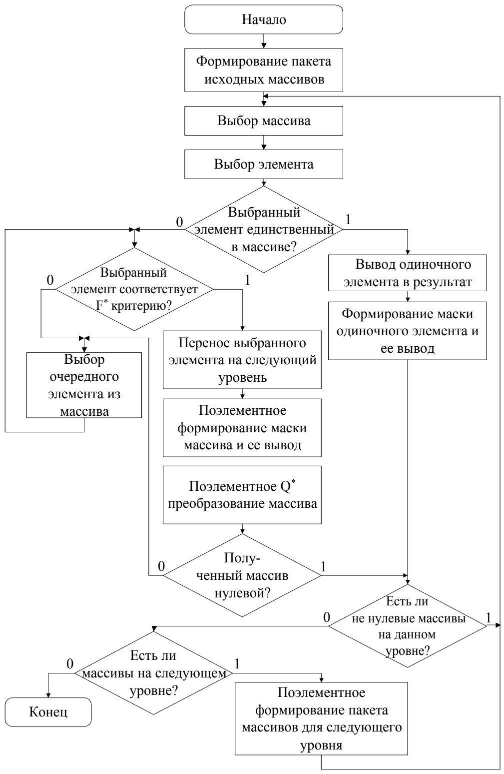


Рисунок 5 – Структурная схема алгоритма поэлементного ПИ преобразования

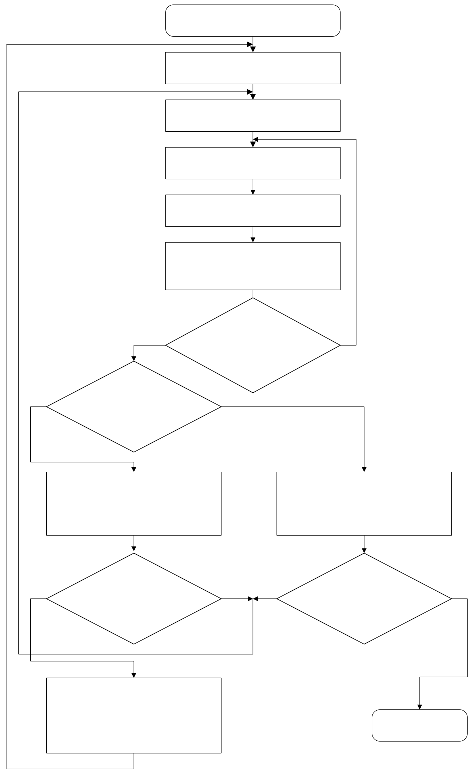


Рисунок 6 – Структурная схема обратного поэлементного ПИ преобразования

Выводы

Разработанные формы параллелизма для параллельно-иерархических структур четырех классов – ОКОД, ОКМД, МКОД, МКМД, позволяют их использовать для решения различных прикладных задач, для работы которых требуется разный масштаб времени.

Литература

1. Прангишвили И.В. Параллельные вычислительные системы с общим управлением / [Прангишвили И.В. и др.]. – М. : Энергоатомиздат, 1983. – 312 с.
2. Flynn H.J. Some Computer Organizations and Their Effectiveness / H.J. Flynn // IEEE Trans. Comput. – C 21.
3. Карцев М.А. Вычислительные системы и синхронная арифметика / М.А. Карцев, В.А. Брик. – М. : Радио и связь, 1981.

L.I. Tymchenko, I.D. Ivasyuk, R.V. Makarenko

Fundamental Computing Structures and Algorithms on the Base of Parallel-Hierarchical Transformation

The possibility of development of crucially new calculating structures and algorithms is described in the article. For solving this task the theory of parallel-hierarchical transformation is offered which is aimed into achievement the maximum possible algorithmic and schematic speed.

Статья поступила в редакцию 29.06.2010.